

사용자의 동적인 요청을 고려한 소형셀 캐시 전략

김재관*, 이수경°, 김예나*

Small Cell Cache Strategy Considering User's Dynamic Request

JaeKwan Kim*, SuKyoung Lee°, YeNa Kim*

요약

소형셀 네트워크에서의 파일 전송 및 캐싱을 위해 기존에 제안된 비용 함수는 사용자의 동적 요청으로 인한 소형셀 캐시 파일의 구성 변경을 고려하지 않았다. 본 논문에서는 이러한 구성 변경을 고려한 비용 모델과 이에 최적화된 캐시 전략을 제안한다. 시뮬레이션을 통해 제안한 비용 모델 기반의 캐시 전략이 기존의 캐시 전략 보다 효율적임을 확인하였다.

Key Words : Small cell network, Cache Strategy, Dynamic request, Cost optimization, Popularity parameter

ABSTRACT

Regarding file transfer and caching in a small cell network, the existing cost function does not consider the change in the configuration of the small cell cache file due to user's dynamic request. In this paper, we propose a reasonable cost model that adds these changes to the existing cost model and the optimized cache strategy. Simulation results show the effectiveness of the proposed cache strategy comparing the existing cache strategy.

I. 서론

소형셀 캐시 기술 연구는 2020년 상용화를 목표로 하고 있는 5G 네트워크에서 기하급수적으로 증가하는 데이터 트래픽을 효율적으로 처리하기 위한 중요한 연구 분야중 하나이다^[1].

소형셀 캐시에 관한 기존의 연구로서 [2,3]은 제안한 비용 함수의 사용자 요청에 대한 기댓값을 최소화하는 방향으로 캐시 전략을 초기에 결정하였다. 단, 사용자의 요청 및 기지국의 파일 전송이 진행 중일 때는 소형셀에 캐시된 파일 삭제 및 추가가 불가능하다.

특히, [4]는 소형셀 네트워크에서 최적의 캐시 전략을 위하여 두 가지 요소의 선형적인 합에 의존한 비용 함수를 제시하였다. 첫 번째는 코어망에 연결된 대형셀이 사용자들에게 전송하는 파일의 양이고, 두 번째는 대형셀이 전송하는 파일의 양을 줄이기 위해 캐시한 파일의 양이다.

그러나, 사용자의 동적인 요청에 응답하는 실제 소형셀 네트워크 환경에서는 소형셀 캐시 구성의 변화량에 대한 요소도 비용으로 함께 고려되어야 한다. 즉, 시스템의 전체적인 성능을 정확히 반영하기 위해, 캐시 구성의 변화에 이용된 백홀(backhaul) 부하 및 이에 대한 소형셀 내부의 부하를 비용으로 포함시켜야 한다.

따라서, 본 논문에서는 캐시 구성의 변화에 대한 요소를 포함한 세 가지 중요한 요소를 최적화하는 캐시 전략을 제안한다.

II. 시스템 모델

본 논문에서 고려한 소형셀 네트워크는 대형셀 1개, 소형셀 K 개로 구성된다. 대형셀은 각 소형셀과 백홀 링크로 연결되어 있다^[2-4]. 대형셀은 코어망에 저장된 파일을 소형셀에 전송하는 역할을, 소형셀은 내부에 캐시된 파일 혹은 대형셀로부터 전송받은 파일을 사용자들에게 전송하는 역할을 한다^[3]. 대형셀과 연결된 코어망에 저장된 파일의 수는 모두 N 개이며, 각 파일의 크기는 모두 같다고 가정한다. 각 소형셀과 연결된 사용자들의 수는 M 명이다. 각 사용자는 인기도 파라미

* 본 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No.NRF-2017R1A2B4002000)

• First Author : (ORCID:0000-0002-9466-8663)Yonsei University Department of Computer Science College of Engineering, lotusstar@yonsei.ac.kr, 학생회원

° Corresponding Author : Yonsei University Department of Computer Science College of Engineering, sklee@yonsei.ac.kr, 종신회원

* National Institute of Standards and Technology, yena.kim@nist.gov

논문번호 : 201807-225-B-LU, Received July 24, 2018; Revised September 10, 2018; Accepted September 10, 2018

터 γ 에 기반한 확률에 따라 파일을 요청한다^[2-4]. 즉, n 번째($n = 1, 2, \dots, N$) 파일을 요청할 확률은

$$p_n = \frac{n^{-\gamma}}{\sum_{j=1}^N j^{-\gamma}} \text{이다.}$$

다음으로, 사용자의 동적인 요청을 나타내기 위해 시간 변수 t 를 도입하였다($t = 1, 2, 3, \dots$). 이에 따라 사용자의 동적인 요청은 t 에 대한 함수인 $r_{ki}(t) \in \{1, 2, \dots, N\}$ 로 나타낸다. $r_{ki}(t)$ 는 t 에 소형셀 $k(k = 1, 2, \dots, K)$ 의 i 번째($i = 1, 2, \dots, I$) 사용자가 p_n 의 확률 분포에 따라 요청한 파일의 순서이다. 즉, n 번째 파일을 요청하면 $r_{kn}(t) = n$ 이다. 이를 소형셀 관점으로 정리하면, k 번째 소형셀에서 n 번째 파일을 요청한 사용자의 수 $R_{kn}(t)$ 는 식 (1)과 같다.

$$R_{kn}(t) = \sum_{i=1}^I [r_{ki}(t) = n] \quad (1)$$

또한, 본 논문에서는 단위 시간 프레임 내에서 사용자들의 요청 $R_{kn}(t)$ 를 받은 직후에 4장과 같이 캐시 구성 $C_{kn}(t) \in \{0, 1\}$ 가 결정됨을 가정한다. 여기서, k 번째 소형셀이 n 번째 파일을 캐시할 경우 $C_{kn}(t) = 1$, 그렇지 않으면 $C_{kn}(t) = 0$ 이다. 이를 행

렬로 표현하면 $C_{K \times N}(t) = \begin{bmatrix} C_{11}(t) & C_{12}(t) & \dots & C_{1N}(t) \\ C_{21}(t) & C_{22}(t) & \dots & C_{2N}(t) \\ \vdots & \vdots & \ddots & \vdots \\ C_{K1}(t) & C_{K2}(t) & \dots & C_{KN}(t) \end{bmatrix}$ 이다.

III. 네트워크 비용 함수

본 장에서는 기존의 비용 함수와 제안하는 비용 함수에 대해서 설명한다.

3.1 기존의 비용 함수

2장의 시스템 모델을 기반으로 백홀 부하의 양과 캐시 사이즈의 합으로 계산하는 기존의 비용 함수^[4] $L_2(t)$ 는 식 (2)와 같다.

$$L_2(t) = \sum_{k=1}^K \sum_{n=1}^N [R_{kn}(t) \{1 - C_{kn}(t)\} + cC_{kn}(t)] \quad (2)$$

소형셀 k 와 n 번째 파일에 대하여, $C_{kn}(t) = 0$ 일 때는 해당 파일을 요청한 사용자들의 수 $R_{kn}(t)$ 만큼 백홀 부하의 양에 대한 비용이 발생한다. 반면에, $C_{kn}(t) = 1$ 일 때는 백홀 트래픽이 발생하지 않지만, 파

일을 캐시하는 비용 c 가 발생한다. 이는 미들박스(middlebox)와 같이 기지국에 설치된 클라우드(cloud)에서의 의해 발생하는 저장 및 계산 비용이다^[5].

3.2 제안하는 비용 함수

본 논문에서는 소형셀 네트워크에서 시스템의 전체적인 성능을 정확히 반영하기 위해 c 에 더하여 새로운 비용 함수를 제시하였다. 즉, (2)의 비용 함수에 캐시 구성의 변화에 대한 요소가 추가되었다. 새로 제안하는 비용 함수 $L_3(t)$ 는 식 (2)의 두 가지 요소와 새로 추가된 요소를 포함한 세 가지 요소의 합으로 다음 식 (3)과 같이 나타낸다.

$$L_3(t) = \sum_{k=1}^K \sum_{n=1}^N \left[R_{kn}(t) \{1 - C_{kn}(t)\} + cC_{kn}(t) + dC_{kn}(t) \{1 - C_{kn}(t-1)\} \right] \quad (3)$$

여기서 새로 추가된 비용은 이전 시간 프레임($t-1$)에 캐시되지 않은 파일이 현재 시간 프레임(t)에 캐시될 때만 발생한다고 가정하였다. 즉, 소형셀 k 와 n 번째 파일에 대해 $C_{kn}(t-1) = 0$ 이고 $C_{kn}(t) = 1$ 일 때만 비용 d 가 발생한다. 여기서 d 는 캐시 변화에 필요한 백홀의 양(1)과 이로 인해 발생하는 클라우드의 계산 작업에 대한 비용($d-1$)이 통합된 비례 상수이다. d 는 최소 1이며, 클라우드의 사용 빈도가 증가함에 따라 그 값이 커질 수 있다. 캐시에 저장된 파일의 구성이 수정되지 않는 경우^[3] d 를 무한으로 설정하는 것과 같은 효과를 보인다.

IV. 최적의 캐시 전략

식 (2)의 기존의 비용 함수 $L_2(t)$ 를 최소로 하는 최적의 캐시 전략은 식 (4)와 같이 각 소형셀에서 c 보다 많은 수의 사용자가 요청한 파일을 캐시하는 방법이다.

$$C_{K \times N}(t) = \arg \min_{C_{k \times n}(t)} L_2(t) \quad (4-1)$$

$$C_{kn}(t) = \begin{cases} 1 \cdots R_{kn}(t) > c \\ 0 \cdots \text{otherwise} \end{cases} \quad (4-2)$$

다음으로, 식 (3)의 제안하는 비용 함수 $L_3(t)$ 를 최소로 하는 최적의 캐시 전략은 식 (5)와 같이 각 소형셀에서 이전 시간 프레임에 캐시된 파일을 요청한 사용자의 수가 c 보다 클 때, 이전 시간 프레임에 캐시되지 않은 파일을 요청한 사용자의 수가 $(c+d)$ 보다

클 때 그 파일을 캐시하는 방법이다.

$$C_{K \times N}(t) = \arg \min_{c_{k,n}(t)} L_3(t) \quad (5-1)$$

$$C_{k_n}(t) = \begin{cases} 1 \cdots R_{k_n}(t) > \begin{cases} c & \cdots C_{k_n}(t-1) = 1 \\ c+d & \cdots C_{k_n}(t-1) = 0 \end{cases} \\ 0 \cdots \text{otherwise} \end{cases} \quad (5-2)$$

제안한 캐시 전략은 백홀 부하의 양, 캐시 사이즈, 그리고 캐시 구성의 변화 간에 트레이드오프를 이루게 함으로써 사용자의 동적인 요청을 고려한 소형셀 네트워크의 성능을 향상시킬 수 있다.

V. 시뮬레이션 분석

본 논문에 대한 시뮬레이션 환경으로 이기종 소형셀 네트워크^[3]에서 대형셀 유니캐스트 전송방식^[4] 및 인기도 파라미터 기반의 파일 요청 모델^[3,4]을 이용하였다.

제안한 비용 함수 (3)에 두 가지(기존, 제안)의 캐시 전략 (4), (5)를 각각 대입하여 시뮬레이션을 수행하였다. 표 1은 시스템 파라미터들을 보여주며, d 는 1 ~ 5의 독립 변수로 설정하였다.

그림 1, 2는 d 가 3일 때 시간 프레임(t) 50단위 동안 발생한 비용에 대한 그래프를 보여준다. 본 그래프에서의 비용은 모든 소형셀이 파일을 캐시하지 않을 때의 비용을 기준(1)으로 표준화하였다. 비용의 종류는 기존의 비용 (2), 새로 추가된 비용(캐시 구성의 변화), 그리고 제안하는 전체(기존+추가) 비용 (3)의 3가지이며, 캐시 전략은 기존 및 제안한 캐시 전략 (5), (6)의 2가지로, 그래프에서는 $3 \times 2 = 6$ 가지의 실선(기존 캐시전략) 및 점선(제안 캐시전략)을 통하여 비용을 표현하였다. 그림 1은 시뮬레이션을 1회 시행하였을 때의 비용(예시)을, 그림 2는 100회 시행하여 평균한 비용(예시)을 나타낸다.

그림 3~5는 그림 2의 비용 함수들을 t 에 대하여 평균한 값을 d 에 따라 나타낸 것이다. 그림 3은 기존의 비용, 그림 4는 새로 추가된 비용, 그림 5는 제안하는

표 1. 시스템 파라미터
Table 1. System Parameter

Parameter	Value	Parameter	Value
K	10	γ	1.0
N	10	c	3.5
I	30		

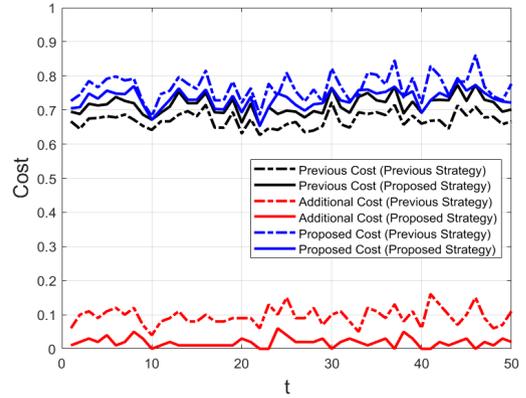


그림 1. 시간 프레임에 따른 비용 (1회 시행)
Fig. 1. Time frame vs Cost (Once)

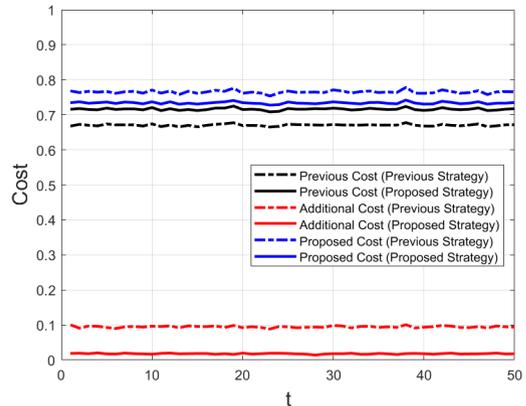


그림 2. 시간 프레임에 따른 비용 (100회 시행 평균)
Fig. 2. Time frame vs Cost (100 times average)

전체 비용을 나타내었다. d 가 증가할수록 캐시 구성의 변화에 민감한 비용 함수가 되므로, 제안된 캐시 전략은 그림 4와 같이 캐시 구성의 변화에 대한 비용을 크게 줄일 수 있다. 대신에, 트레이드오프 관계에 따라 그림 3과 같이 기존에 제시된 비용이 커지게 된다. 특히, 제안하는 캐시 전략의 성능(빨간색 실선)이 계단 모양으로 증가하는 이유는 식 (5)의 캐시 전략이 $(c+d)$ 의 정수 부분에 의존하기 때문이다. 최종적으로, 그림 5와 같이 제안하는 캐시 비용이 결정되며 전반적으로 d 가 증가할수록 제안하는 캐시 전략이 비용 효율적이다.

VI. 결론

본 논문에서는 소형셀 캐시 네트워크에서 사용자의 동적인 요청을 고려한 새로운 캐시 전략을 제안하였다. 사용자들로의 파일 전송량을 제외하고, 소형셀에 캐시된 파일의 양만으로는 동적으로 변하는 캐시 파일 구성에 대한 비용이 고려되지 않아 정확한 캐시 전략을 구할 수 없으므로, 본 논문에서는 이를 고려한 새로운 비용 함수를 제시하였다. 시뮬레이션을 통해, 제안한 비용 모델 기반의 캐시 전략이 기존의 캐시 전략보다 효율적임을 확인하였다.

References

- [1] H. K. Jung, S. Y. Jung, D. H. Lee, S. Q. Lee, and J. H. Kim, "Wireless caching algorithm based on user's context in smallcell environments," *J. KICS*, vol. 40, no. 7, pp. 789-798, Jul. 2016.
- [2] V. Bioglio, F. Gabry, and I. Land, "Optimizing MDS codes for caching at the edge," *2015 IEEE GLOBECOM*, San Diego, CA, USA, Dec. 2015.
- [3] J. Liao, K. K. Wong, M. R. A. Khandaker, and Z. Zheng, "Optimizing cache placement for heterogeneous small cell networks," *IEEE Commun. Lett.*, vol. 21, no. 1, pp. 120-123, Jan. 2017.
- [4] A. Gharaibeh, A. Khreishah, B. Ji, and M. Ayyash, "A provably efficient online collaborative caching algorithm for multicell-coordinated systems," *IEEE Trans. Mob. Comput.*, vol. 15, no. 8, pp. 1863-1876, Aug. 2016.
- [5] J. Sherry, S. Hasan, C. Scott, A. Krishnamurthy, S. Ratnasamy, and V. Sekar, "Making middleboxes someone else's problem: network processing as a cloud service," *ACM SIGCOMM Computer Commun. Rev.*, vol. 42, no. 4, pp. 13-24, Apr. 2012.

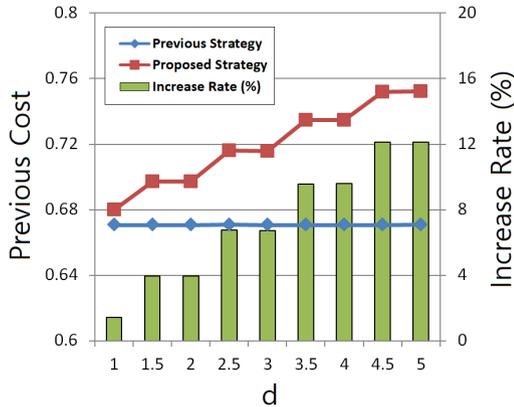


그림 3. 기존 비용
Fig. 3. Previous Cost

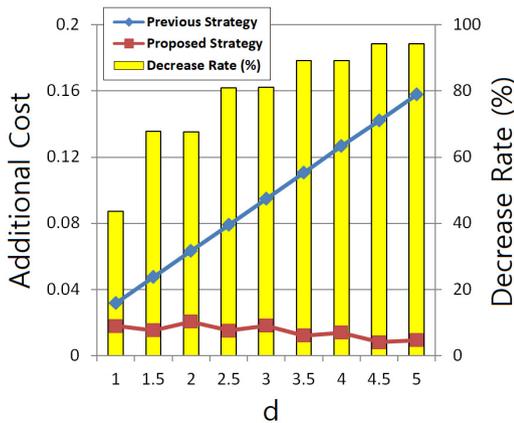


그림 4. 추가 비용
Fig. 4. Additional Cost

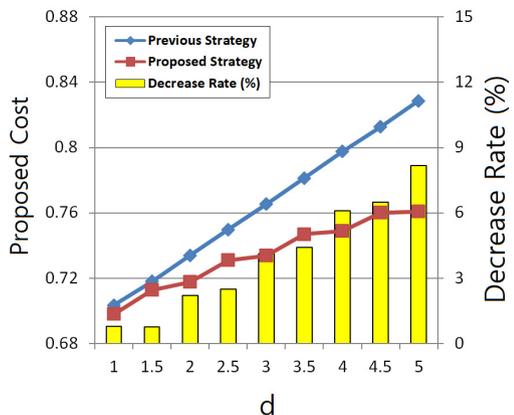


그림 5. 제안 비용
Fig. 5. Proposed Cost